

(51) Int.Cl. ⁷	識別記号	F I	テ-マコ-ト(参考)
H 04 N 5/44		H 04 N 5/44	Z 5 B 0 6 9
G 06 F 3/14	3 1 0	C 06 F 3/14	3 1 0 C 5 C 0 2 6
H 04 H 1/00		H 04 H 1/00	C 5 C 0 6 4
H 04 N 5/445		H 04 N 5/445	Z
5/46		5/46	

審査請求 未請求 請求項の数22 OL (全 25 頁) 最終頁に続く

(21)出願番号	特願2001-287056(P2001-287056)	(71)出願人 000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22)出願日	平成13年9月20日(2001.9.20)	(72)発明者 山本 高司 東京都大田区下丸子3丁目30番2号キヤノン株式会社内
		(72)発明者 荒谷 俊太郎 東京都大田区下丸子3丁目30番2号キヤノン株式会社内
		(74)代理人 100090538 弁理士 西山 恵三 (外1名)
		最終頁に続く

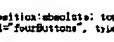
(54)【発明の名称】受信システム

(57)【要約】

【課題】 簡単な操作にて複数の番組を同時に視聴可能とする。

【解決手段】 受信システムは、複数のテレビジョン放送システムの番組放送予定情報に基づき、複数の受信すべき放送チャンネルを指定すると共に前記複数の放送チャンネルによる放送番組の映像の表示レイアウトを制御する制御情報を生成して配信する配信装置と、前記配信装置により配信された制御情報を入力し、前記制御情報にて指定された複数の受信チャンネルのテレビジョン放送信号を同時に受信すると共に、これら複数のチャンネルのテレビジョン放送番組の映像を前記制御情報に従うレイアウトにて表示する受信装置とを有する構成とした。

```

<body onloaded="initialize();>
<div id="buttonRed" type="button">selectRed0</div>
<div id="buttonGreen" type="button">selectGreen0</div>
<div id="buttonBlue" type="button">selectBlue0</div>
<div id="buttonYellow" type="button">selectYellow0</div>
<div id="div_video0" type="div">
<object id="video0", type="video/mpeg2", source=source0></object>
</div>
<div id="div_video1" type="div">
<object id="video1", type="video/mpeg2", source=source1></object>
</div>
<div id="div_video2" type="div">
<object id="video2", type="video/mpeg2", source=source2></object>
</div>
<div id="div_video3" type="div">
<object id="video3", type="video/mpeg2", source=source3></object>
</div>
<div id="div_audio0" type="div">
<object id="audio0", type="audio/mpeg2", source=source0></object>
</div>
<div id="div_audio1" type="div">
<object id="audio1", type="audio/mpeg2", source=source1></object>
</div>
<div id="div_audio2" type="div">
<object id="audio2", type="audio/mpeg2", source=source2></object>
</div>
<div id="div_audio3" type="div">
<object id="audio3", type="audio/mpeg2", source=source3></object>
</div>
<div style="position: absolute; top: 580; left: 60; width: 960; height: 50;">
<object id="advertisement0", type="image/jpeg" data="adv.jpg"></object>
</div>
<div style="position: absolute; top: 680; left: 60; width: 300; height: 50;">
<object id="fourButtons", type="image/jpeg" data="fourButtons.jpg"></object>
</div>
<div style="position: absolute; top: 680; left: 360; width: 600; height: 60;">

</div>
<div id="div_bttnR">
<object id="buttonRed", type="image/jpeg" data="bttnR.jpg"></object>
</div>
<div id="div_bttnG">
<object id="buttonGreen", type="image/jpeg" data="bttnG.jpg"></object>
</div>
<div id="div_bttnB">
<object id="buttonBlue", type="image/jpeg" data="bttnB.jpg"></object>
</div>
<div id="div_bttnY">
<object id="buttonYellow", type="image/jpeg" data="bttnY.jpg"></object>
</div>
</body>

```

【特許請求の範囲】

【請求項1】 複数のテレビジョン放送システムの番組放送予定情報に基づき、複数の受信すべき放送チャンネルを指定すると共に前記複数の放送チャンネルによる放送番組の映像の表示レイアウトを制御する制御情報を生成して配信する配信装置と、

前記配信装置により配信された制御情報を入力し、前記制御情報にて指定された複数の受信チャンネルのテレビジョン放送信号を同時に受信すると共に、これら複数のチャンネルのテレビジョン放送番組の映像を前記制御情報に従うレイアウトにて表示する受信装置とを有する受信システム。

【請求項2】 前記配信装置は前記複数のテレビジョン放送システムのうちの一つが放送する放送チャンネルにより前記制御情報を配信することを特徴とする請求項1記載の受信システム。

【請求項3】 前記配信装置は前記複数のテレビジョン放送システム以外の放送システムは放送する放送チャンネルにより前記制御情報を配信することを特徴とする請求項1記載の受信システム。

【請求項4】 前記制御情報はXMLによって記述されたデータであり、前記配信装置はBSデジタルテレビジョン放送のデータ放送チャンネルにより前記制御情報を配信することを特徴とする請求項2または3記載の受信システム。

【請求項5】 前記受信装置は前記受信装置の性能に係る性能情報を前記配信装置に対して送信し、前記配信装置は更に、前記番組放送予定情報及び前記性能情報とに基づいて前記制御情報を生成することを特徴とする請求項1記載の受信システム。

【請求項6】 前記性能情報は、前記受信装置が同時に受信処理可能なテレビジョン放送チャンネルの数、表示部の表示解像度、前記受信装置が同時に出力可能なテレビジョン放送番組に係る音声の数のうちの少なくとも一つを含むことを特徴とする請求項5記載の受信システム。

【請求項7】 前記配信手段は前記テレビジョン放送チャンネルとは異なる配信路を介して前記制御情報を配信することを特徴とする請求項1記載の受信システム。

【請求項8】 前記配信手段はインターネットを介して前記制御情報を配信し、前記受信手段は前記インターネットを介して前記制御情報をダウンロードすることを特徴とする請求項7記載の受信システム。

【請求項9】 前記配信装置は更に、前記複数の放送チャンネルによる放送番組の映像と共に表示すべき付加画像データと前記付加画像データの表示レイアウトとを前記制御情報として生成することを特徴とする請求項1記載の受信システム。

【請求項10】 前記配信装置は、前記指定した放送チャンネルの内容に基づいて前記付加画像データを生成す

ると共に前記付加画像データの表示レイアウトを決定することを特徴とする請求項9記載の受信システム。

【請求項11】 前記受信装置はユーザプロファイル情報を前記配信装置に対して送信し、前記配信装置は更に、前記番組放送予定情報及び前記ユーザプロファイル情報とに基づいて前記制御情報を生成することを特徴とする請求項1記載の受信システム。

【請求項12】 前記ユーザプロファイル情報はテレビジョン放送番組の視聴時間に係る情報を含むことを特徴とする請求項1記載の受信システム。

【請求項13】 前記ユーザプロファイル情報はテレビジョン放送番組の内容に関する情報を含むことを特徴とする請求項1記載の受信システム。

【請求項14】 前記ユーザプロファイル情報は視聴目的に関する情報を含むことを特徴とする請求項1記載の受信システム。

【請求項15】 テレビジョン放送信号を受信する受信装置と共に用いられ、複数のテレビジョン放送システムの番組放送予定情報に基づき、複数の受信すべき放送チャンネルを指定すると共に前記複数の放送チャンネルによる放送番組の映像の表示レイアウトを制御する制御情報を生成して前記受信装置に対して配信する配信装置であって、

前記受信装置において前記制御情報にて指定された複数の受信チャンネルのテレビジョン放送信号を同時に受信すると共に、これら複数のチャンネルのテレビジョン放送番組の映像を前記制御情報に従うレイアウトにて表示するようになされたことを特徴とする配信装置。

【請求項16】 テレビジョン放送波を受信し、映像信号を出力する複数の受信手段と、

前記複数の受信手段により受信された複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する映像処理手段と、配信装置により複数のテレビジョン放送システムの番組放送予定情報に基づいて生成されて配信される情報であって、複数の受信すべき放送チャンネルを指定すると共に前記複数の放送チャンネルによる放送番組の映像の表示レイアウトを制御する制御情報を入力する入力手段と、

前記制御情報に従って前記複数の受信手段と前記映像処理手段とを制御し、前記複数の受信手段により受信すべき放送チャンネルを自動的に指定すると共に前記映像処理手段による表示レイアウトとを自動的に設定する制御手段とを備える受信装置。

【請求項17】 前記受信装置の性能に係る性能情報を前記配信装置に対して送信するインターフェイス手段を備え、前記配信装置は前記番組放送予定情報及び前記性能情報をとに基づいて前記制御情報を生成することを特徴とする請求項16記載の受信装置。

【請求項18】 前記性能情報は、前記受信装置が同時

に受信処理可能なテレビジョン放送チャンネルの数、表示部の表示解像度、前記受信装置が同時に出力可能なテレビジョン放送番組に係る音声の数のうちの少なくとも一つを含むことを特徴とする請求項17記載の受信装置。

【請求項19】ユーザプロファイル情報を前記配信装置に対して送信するインターフェイス手段を備え、前記配信装置は前記番組放送予定情報及び前記ユーザプロファイル情報に基づいて前記制御情報を生成することを特徴とする請求項16記載の受信装置。

【請求項20】前記ユーザプロファイル情報はテレビジョン放送番組の視聴時間に係る情報を含むことを特徴とする請求項19記載の受信装置。

【請求項21】前記ユーザプロファイル情報はテレビジョン放送番組の内容に関する情報を含むことを特徴とする請求項19記載の受信装置。

【請求項22】前記ユーザプロファイル情報は視聴目的に関する情報を含むことを特徴とする請求項19記載の受信装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は受信システムに関し、特に、複数のチャンネルの映像を同時に受信して表示するシステムに関する。

【0002】

【従来の技術】従来、テレビ受信機において二台のチューナを搭載するなどして二つの放送番組を受信して同時に画面に表示する機能が知られている。このような機能により、ある放送番組を視聴しながら同じ時間帯に重なった別の放送番組も視聴することができる。

【0003】一方、今日、ケーブルテレビジョン(CATV)、CS放送に加えてBSデジタル放送の開始により、従来とは比較にならない程多くのチャンネルにより放送が行われている。

【0004】また、BSデジタル放送においては複数の映像により一つの放送番組を構成するマルチビュー放送が行われる。これを自由に視聴できるようテレビ受信機側も複数のチューナやデコーダを備え、より多くの画面を同時に表示する機能を持つことが考えられる。こうした場合、テレビ受信機の表示画面のレイアウトも、従来のようにテレビ受信機に予め用意しているレイアウトだけではなく、より自由なレイアウトが望まれるようになる。

【0005】

【発明が解決しようとする課題】このように多くのチャンネルにより非常に多数番組が放送されるようになると、テレビ視聴の目的に適う番組がどの時間帯でどのチャンネルで放送されるかを正確に把握することが難しくなる。ユーザは視聴の目的に適う番組を探すため、従来から提供される雑誌、新聞に掲載される番組予定表を精

査する事に加え、BSデジタル放送においては放送波に多重されている電子プログラムガイドを利用して番組予定も調べなければならない。さらにこれら多数の番組の中から同時に視聴すべき番組の組み合わせを探し出して視聴計画を立案するには、多大な労力と時間を要する。娯楽としてのテレビ視聴にそぐわない状況である。

【0006】また、テレビ受信機の機能が高度化してより多くの画面を同時に表示できるようになり、表示レイアウトの柔軟性を備えてくると、このようなテレビ受信機の機能の多さに比例して複雑な操作を要するようになる。

【0007】更に、視聴する番組の選択操作まで考慮すると快適な操作は困難な状況である。

【0008】本発明はこのような問題点を解決し、簡単な操作にて複数の番組を同時に視聴可能とすること目的とする。

【0009】本発明の他の目的は、多数のチャンネルの中から視聴すべき番組を選択する操作や、複数の番組を同時に視聴する際の表示レイアウトの設定操作などの負担を軽減する処にある。

【0010】

【課題を解決するための手段】前記課題を解決するため、本発明においては、複数のチャンネルを有するテレビジョン放送波を受信し、映像信号及び制御信号を出力する複数の受信手段と、前記複数の受信手段により受信された複数の映像信号に係る映像を同一画面上に表示すべく前記複数の映像信号を合成処理して表示部に出力する映像処理手段と、前記受信手段より出力された制御信号に従って前記複数の受信手段と前記映像処理手段とを制御し、前記複数の受信手段により受信すべきチャンネルと前記映像処理手段による表示レイアウトとを自動的に設定する制御手段とを備える構成とした。

【0011】

【発明の実施の形態】以下、図面を参照して本発明の実施形態について説明する。

【0012】図1は本発明が適用されるテレビ受信機100を含む視聴システムの構成を示した図である。

【0013】図1において、101は衛星放送受信アンテナ、102、103、104はアンテナにより受信された信号から指定されたチャンネルを選局し、検波、およびデスクランブルを行ってMPEG2-トランスポートストリーム(以下MPEG2-TS)を出力する機能を含むBSデジタルチューナ、105、106、107はチューナ102、103、104から出力されたMPEG2-TSから番組ごとに指定されたビデオストリーム、オーディオストリーム、データ放送データ、および番組配列情報等を抽出するトランスポートデコーダ、108、109はビデオストリームを復号化するビデオデコーダ、110はオーディオストリームを復号化するオーディオデコーダ、111、112は復号化されたビデ

オデータをフレームごとに格納するビデオメモリ、113は復号化されたオーディオ信号を出力するスピーカ、114はデータ放送画面や画面操作のために文字や図形などの表示データを生成するグラフィック生成部、115はデコードされたビデオ信号による映像と、グラフィック生成部114からのグラフィックデータによる映像を一つの表示上に表示するよう合成処理する合成プロセッサ、116は画面表示を行う表示部、117はリモコンと操作ボタンを含む操作入力部、118はユーザからの操作入出力と番組配列情報の入力に基づいてデジタルテレビ受信機100の制御を行うと共に、データ放送データ入力によりグラフィック生成部114を通じてデータ放送表示画面を描画する制御部であり、100はこうした要素から構成されるデジタルテレビ受信機である。

【0014】また、図1の121、122はBSデジタル放送の放送局であり、データ放送を含む放送番組を放送衛星124を通じて配信する。123はBSデジタル放送データ放送の放送局でありデータ放送番組を放送衛星124を通じて配信する。放送局123が放送するデータ放送番組の一部は、これから説明するマルチ画面テンプレート機能を果す番組である。データ放送の放送局123は、放送局121、122が放送する放送番組の放送予定を入手し、これらの放送の選局制御と受信機のマルチ画面機能制御を行うスクリプトを含むXMLデータをデータ放送チャンネルにて配信する。

【0015】放送局321または322から配信される放送番組はアンテナ301を経てチューナ302または303で受信される。チューナ102によって得られたMPEG2-TSトランスポートデコーダ105に出力され、ここでビデオストリーム、オーディオストリーム、データストリームが抽出される。ビデオストリームはビデオデコーダ108によってビデオ信号に復号化され、フレームごとにビデオメモリ111に格納され、合成プロセッサ115を通して表示部116に表示される。

【0016】また、オーディオストリームはオーディオデコーダ110でオーディオ信号に復号化され、スピーカ113から音声として出力される。こうして一つの放送番組が受信され視聴者に提示される。

【0017】また、チューナ103からトランスポートデコーダ106、ビデオデコーダ109、メモリ112を通じて別のチャンネル放送番組を並行して受信することができる。これら二つのビデオ信号は合成プロセッサで合成され、表示部116に同時に表示される。オーディオデータはユーザの操作によってどちらか一方の信号が選択され、スピーカ113から出力される。

【0018】一方、放送局123から放送衛星124を経て配信されるデータ放送は、ISO/IEC13818-6に規定されるDSM-CCのデータカルーセル方式により放送局123から繰り返し送出される放送番組であり、チューナ1

04によって受信されて、トランスポートデコーダ107によってデータストリームが抽出され制御部118へ送られる。

【0019】データストリームにはテキスト情報、スクリプト情報、画像情報、映像音声情報が含まれ、テキスト情報はW3Cが規定するXML (eXtensible Markup Language) によって記述される。本形態ではデータ放送用XMLの仕様としてHTML4.0をXML1.0で再定式化した XHTML仕様を基本として用いている。XMLでは文書の文字列にタグ（<xxx>と</xxx>に囲まれた部分）によって意味付けされる。また、このタグは「入れ子」構造をとることが可能である。また、表示形式はCSS (Cascading Style Sheet)、スクリプト処理はJavaScriptを用いている。

【0020】こうしたデータは制御部118で解釈、実行され、文字、図形、画像情報がグラフィック生成部114で生成され、合成プロセッサ115を通じて表示部116に表示される。音声情報は図示されないサウンド生成部を通じてスピーカ113から出力される。

【0021】図14は放送局121、122、123から放送される番組の放送予定を表す表である。放送局121はネットワーク“broadcasterA”を使用してHD(HighDefinition)番組をチャンネル101で放送する。また、放送局122はネットワーク“broadcasterB”に含まれるチャンネル161、162、163に各々SD(Standard Definition)番組を1番組ずつ放送するか、3チャンネルを使用してHD番組を放送する。放送局123はネットワーク“broadcasterC”を使用してデータ放送番組をチャンネル421で放送する。

【0022】図2は制御部118に含まれるCPUで動作するソフトウェアの構造を示した図である。データ放送プラウザはオペレーティングシステムの上で動作するアプリケーションであり、アプリケーションからはグラフィックライブラリとグラフィックドライバを介してグラフィック生成部を制御し文字や図形、画像などを合成プロセッサ115に含まれるグラフィックバッファに対して描画する。またウィンドウ制御ミドルウェアと画面合成部ドライバを介して合成プロセッサ115の制御を行い、ビデオデコーダ108、109からメモリ111、112を経由して送られる映像と、静止画、文字図形の合成処理を制御する。また、トランスポートデコーダドライバを介してトランスポートデコーダが制御され、トランスポートストリームに含まれる番組配列情報を受け取る。

【0023】そして、番組配列情報から得られる情報に基づき、チューナドライバ、トランスポートデコードドライバを介してチューナ102、103、104およびトランスポートデコーダ105、106、107を制御して目的の放送番組を構成する映像ストリーム、音声ストリームを獲得し、映像デコーダ1108、109およ

び音声デコーダ110に送る。また、番組配列情報から得られる情報に基づき、チューナ104及びトランスポートデコーダ107を制御し、データ放送データの受信が行われる。さらに操作入出力ドライバを介してリモコンおよび操作ボタンとの通信が行われ、ユーザの操作を入力する。

【0024】(データ放送ブラウザの動作)図3は本実施例におけるデータ放送ブラウザの動作を示したフローチャートである。ユーザがテレビ番組視聴中にリモコンや操作ボタンを用いてデータ放送の受信表示を指示すると、制御部118に含まれるCPUはデータ放送受信再生プログラム(ブラウザ)の実行を開始する(S301)。

【0025】起動されたブラウザでは、初めに受信するトランスポートストリームの番組配列情報を取得し(S302)、この情報に基づいてトランスポートデコーダ107を制御してデータカルーセル方式で送信されてくるデータ放送コンテンツの受信を開始して、画面のXMLデータおよび同データに埋め込まれた画像などの関連データを取得し、メモリに格納する(S303)。

【0026】図4に同時視聴を行うデータ放送のための「受信XMLデータ」の例を示す。

【0027】このデータ放送は、番組の動作として、他の複数の番組の選局制御とマルチ画面におけるレイアウト制御を含む。以下、このXMLデータによる複数番組の同時視聴機能をマルチ画面テンプレートの機能という。

【0028】具体的には、図18で示す放送番組の中のチャンネル421の番組「極楽ゴルフ三昧」における時刻14時過ぎのものであり、チャンネル101で放送される番組「女子ゴルフツアー中継」と、「男子プロゴルフツアー中継」を構成するチャンネル161, 162, 163の番組を、受信機のマルチ画面機能を利用して同時視聴しようというものである。番組の企画上では4番組を同時に合成表示し、視聴者が興味をもつ男子、女子のプロゴルフ番組を並行して視聴すると共に、その時々で注目する番組を見やすく拡大して表示する操作を可能とするものである。

【0029】しかしながら、マルチ画面機能を有するテレビ受信機といっても、実際に使用される受信機によっては搭載するチューナの数が異なるなど、視聴者ごとに受信機の機能、性能に差がある。図4に示すデータ放送のXMLデータにおいては、初めに受信機で使用できるチューナの数を調べて、適切な別のXMLデータへ分岐する動作を行う。

【0030】XMLデータの<body>部には本来、放送番組の映像を表示する為のオブジェクトや、音声出力のためのオブジェクト、画面に表示するの文字図形要素等が含まれるが、実際に画面の表示制御を行うのはこのXMLデータから呼び出されたXMLデータなので、ここで

は後で説明するユーザに対する警告表示のための表示オブジェクト401だけが含まれている。<body>部のonLoad属性には、XMLデータが起動されたとき、初めに実行すべきスクリプトがinitialize()であると記述され、スクリプトinitialize()は<head>部の<script>部に定義されている。

【0031】本形態でのブラウザはXMLデータに従って、初めに受信機のチューナの数を調査(S301)するが、これはブラウザの組み込みAPIであるnumberOfTuners()APIで実装されている。numberOfTuners()APIは受信機で使用できるチューナの数をリターンし、初期スクリプトではチューナの数によって分岐が行われて、適切な別のXMLデータを指定して制御を移す。即ち、チューナの数が2の場合には、2台のチューナで動作するデータ放送を実現するXMLデータである「golfParadiseBy2Tuners.xml」が起動され(S307)、チューナの数が3台以上の場合には、2台のチューナで動作するデータ放送を実現するXMLデータである「golfParadiseBy3Tuners.xml」が起動される(S308)。チューナの数が2に満たないときには、マルチ画面を利用する本データ放送のサービスが利用できないので、画面に説明を表示してユーザに通知する(S306)。

【0032】図5、図6、図7は3つのチューナを使用してマルチ画面テンプレート機能を実現するためのXMLデータを示すものである。図5から図7で一つのXMLデータを示している。

【0033】図7の<body>部には、四つの放送番組の映像を表示する為のオブジェクト501や、音声出力のためのオブジェクト502、画面レイアウトを変更するための操作ガイドを表示するボタンの図形要素等503が含まれている。操作ガイドを示す文字や図形要素には表示位置や表示形式を定義するスタイル定義が施されており、例えば501の文字要素には画面の上から680画素、左から360画素のところから、幅500画素、高さ60画素の領域に文字列「ボタンで画面が拡大されます」が配置されるよう属性が与えられている。

【0034】またXMLデータが起動されたとき初めに実行すべきスクリプトがintiallize()であり、リモコンに設けられた専用のボタン(赤、緑、青、黄色)のボタンが操作された場合に実行される実行すべきスクリプトが、各々selectRed()、selectGreen()、selectBlue()、selectYellow()であると記述されている。

【0035】本形態の初期スクリプトでは、データ放送を受信するチューナの他に、二つのチューナを使って四つの放送番組を受信する制御と、これを画面上にレイアウトする制御を実行する。図8はこの制御の手順を示すフローチャートである。

【0036】番組を受信するためには、先ず受信に使用するチューナを割り当てなければならない(図8のS601)。図5の初期スクリプトではこれをopenTuner()A

PIで実現している。openTuner()APIはブラウザが持つ組み込みAPIで、テレビ受信機で使用可能なチューナーに対応したTunerオブジェクトを生成するものである。

【0037】次に番組を受信するために番組局とチャネルの指定を基に番組を選局する。具体的には、ネットワーク名称“broadcasterA”的指定を、データ放送を含むトランスポートストリームのNIT(Network Information Table)情報の中でサーチし(S602)、衛星分配システム情報を獲得する(S603)。ここから得られる周波数などの情報をチューナーに設定して目標チャネルのデータを含むMPEG2-TSを得る(S604)。

【0038】得られたMPEG2-TSをトランスポートデコーダ105でデコードし、指定されたチャネル番号“101”をPAT(Program Association Table)情報の中でサーチして(S605)、対応するPMT(Program Map Table)情報を獲得する(S606)。

【0039】本形態のブラウザでは、これらの制御はTunerオブジェクトが持つaddService()メソッドで実現される。addService()メソッドは、引数に与えられたネットワーク名称とチャネル番号で指定されるサービスの受信をTunerオブジェクトに指示し、serviceオブジェクトを生成して戻り値として返す。serviceオブジェクトはサービスを構成する映像、音声、データなどのコンポーネントを持ち、映像の表示や音声の出力にはこれらのコンポーネントを指定する。

【0040】実際のストリームデータの獲得は、PMT情報から得られる映像ストリーム、音声ストリーム、データストリームを指定するPID(Packet Identifier)を用いて、トランスポートポートデコーダにおけるトランスポートストリームのフィルタリングによって実行される。

【0041】初期スクリプトでは、四つの指定された放送の受信に加え、これらの映像表示と同時に表示するボタン画像の位置制御を行っている。即ちsetPositionX(0)関数を呼び、serviceXオブジェクト(Xは表示位置に対応付けられた0~3の数字)とボタンオブジェクトのstyleに表示位置を設定する。表示位置は、serviceが指定する映像オブジェクトのアスペクト比を考慮した位置となっている。また、serviceオブジェクトのsetPosition0()においては可視の属性と、serviceオブジェクトに属するaudioオブジェクトには音声の出力を指定する属性が指定される。図5においては明示されないsetPosition1()からsetPosition3()においては、audioオブジェクトに音声出力を停止する属性が指定される。

【0042】ブラウザは初期スクリプトの実行後、XMLデータの<body>部分(図5の<body>～</body>部分)を実行する。初めに、オブジェクトとして置かれた四つの映像を表示する。

【0043】即ち、初期スクリプトで指定された放送番組を受信して得られるストリームと、その表示位置として指定されたオブジェクトを、合成プロセッサを制御してディスプレイに合成表示する。また、指定されたストリームによる音声を、音声デコーダを制御してスピーカに出力し、他の描画要素をグラフィック生成部を制御して描画し、合成プロセッサで合成する(S305)。

【0044】図9はこのXMLデータに従う表示画面の例である。

【0045】図9の701は動画ウィンドウとしてXMLデータで指定された101チャネルの放送番組の映像が表示されている。また、702, 703, 704も動画ウィンドウとしてXMLデータで指定された161, 162, 163チャネルの放送番組の映像が表示されている。

【0046】706, 707, 708, 709にはリモコンに設けられた赤、緑、青、黄色ボタンを表す図形が表示され、710, 711の図形と文字による説明表示にある通り、リモコンに設けられた4色のボタンのうちの対応するボタンを押したときに対応する放送番組の映像の表示が左側の大きな画面に移動する事を表現している。712はこのデータ放送番組独自の表示であり、例えば、こうした複数の放送番組を束ねたデータ放送番組企画の内容に沿った広告表示である。

【0047】今、図9の表示状態においてリモコンに設けられた緑色ボタンを押したとき、操作入出力117を通じてブラウザのbodyオブジェクトにonButtonGreenイベントが送られ、XMLデータのselectGreen()スクリプトが実行される。図5に示す通り、selectGreen()スクリプトでは初めにremove1()スクリプトによって画面左に置かれている映像の表示位置を移動し、音声出力を停止する。移動先は、緑色ボタンに対応付けられている161チャネルの映像(service1オブジェクト)の表示位置である。次にservice1オブジェクトの表示位置を画面左の大きな領域に指定し、音声出力を開始する。

【0048】スクリプト実行後の表示状態を図10に示す。図9において右上に表示されていた161チャネルの映像が画面左中央に大きく表示され、代わりに101チャネルの映像が左上位置に移動されている。即ち、リモコンのカラーボタンの操作により、ボタンごとに対応付けられた放送番組の映像を左中央の大きな表示領域に移すことが可能となる。この機能はテレビ受信機1300の通常の機能ではないもので、データ放送によって配信されるXMLデータによって与えられた特有の機能である。

【0049】図11、図12、図13はテレビ受信機の持つチューナーが2台である場合に起動されるXMLデータであり、前述の通り、最初に実行され受信機のチューナーの数を調査するXMLから起動される。図11～図13で一つのXMLデータを示している。ブラウザで実行

されるこのXMLデータは、1台のチューナでXMLデータを配信するデータ放送を受信し、もう一台で四つの放送を1台のチューナで切り替えて受信し、各々最適なレイアウトで表示するマルチ画面テンプレート機能を果すものである。

【0050】図12、図13のXMLデータの<body>部には、図7における説明と同様に、四つの放送番組の映像を表示するためのオブジェクト、音声出力のためのオブジェクト、画面レイアウトを変更するための操作ガイドを表示する文字要素が含まれている。また、初期スクリプトがinitialize()であり、リモコンの左ボタンが押されたときselectLeft()スクリプトが、また右ボタンが押されたときselectRight()スクリプトが呼ばれることが記述されている。

【0051】初期スクリプトinitialize()においては、放送の受信に使用するチューナをブラウザ組み込みのopenTuner()APIによって割り当てられ、ネットワーク名称“broadcasterA”における101チャンネルが選局される。また、service0～3の四つのオブジェクトに対して不可視、音声出力停止の属性が与えられる。最後に初期画面の制御のためにsetPosition0()スクリプトが引数1を伴って呼ばれ、101チャンネルの表示位置と音声出力に関する属性が与えられる。変数magnifiedは拡大表示されておるサービスを示す値（整数）を保持し、初期スクリプトでは初期値として0が与えられる。

【0052】ここで、setPositionX(s)スクリプトの役割が、図7のスクリプトとは異なる。図7のスクリプトでは「場所Xにサービスsを表示するために必要な属性設定を行う」といった動作を行ったが、図12においては「サービスXを条件sのもとに適切な属性設定を行う」という動作を行う。引数sによる条件とは、s=1のとき拡大表示、s=2のとき通常の（小さめの）表示、s=0のとき表示を行わない、といったものである。

【0053】図14は図11～図13のXMLデータによるデータ放送の初期画面である。即ち、初期スクリプトinitialize()から呼ばれるsetPosition0(1)によって、チャンネル101の放送の拡大表示と、これに伴う広告表示713が行われ、音声が出力される。

【0054】図14の表示画面中央下部に文字情報712で示される通り、XMLデータはリモコンの左右方向ボタンによって表示放送を切り替える操作機能を提供する。この機能はテレビ受信機が本来備える機能ではなく、マルチ画面テンプレートとしてのデータ放送によるXMLデータにより実現する機能である。

【0055】図14の状態でリモコンの右ボタンが押されると、図11～図13に示すXMLデータにおいて<body>にイベントが送られ、その結果スクリプトselectRight()スクリプトが呼ばれる。selectRight()においては、現在の変数magnifiedの値が0なので、switch文の制御によりtuner0に対する選局と制御とともに、四つの

setPositionX()スクリプトが呼ばれる。

【0056】前述の通り、引数に-1が指定された場合にはサービスXの表示は消され、引数に0が指定された場合にはサービスXの表示が通常の（小さめの）大きさで表示され、引数に1が指定された場合にはサービスXの表示が拡大表示される。音声出力は拡大表示された番組のものが表示されるよう制御される。サービス1～3はネットワーク“broadcasterB”によるチャンネル161, 162, 163の放送番組であり、チューナー一台で3番組が受信でき、これら三つの放送番組の映像がservice1～3としてマルチ画面機能により合成表示される。

【0057】図15がスクリプトselectRight()が実行された後の表示状態である。チャンネル161の放送番組が左側に大きく表示され、チャンネル162, 163の放送番組がその左側に合成表示される。また、この3番組の合成表示というデータ放送番組企画に沿った広告表示が行われている。

【0058】図16、図17は、図15の状態からさらにリモコンの右ボタンを押したときに、同様のスクリプトが実行された後の表示状態である。順にチャンネル162、チャンネル163の放送番組が拡大表示され、各々のレイアウトに即した広告表示が行われている。

【0059】図17の表示状態からリモコンの左ボタンを押すことにより、図16、図15、図14の状態へ順に変化してゆき、拡大表示された放送番組の音声が出力される。こうしたリモコンの左右ボタンによる表示切り替えの機能はテレビ受信機100の通常の機能にはないもので、図11～図13によるXMLデータによって実現した特有の機能である。

【0060】以上に説明してきた通り、421チャンネルのマルチ画面テンプレート機能のためのデータ放送番組「極楽ゴルフ三昧」は、チャンネル101, 161, 162, 163で放送される4番組を1画面上に表示するための一つの放送番組としてデータ放送チャンネルにて配信される。

【0061】そして、ユーザは、テレビ受信機100のチャンネルの変更操作によって421チャンネルのデータ放送番組「極楽ゴルフ三昧」を選局するだけで、ゴルフに関連した番組を自動的に選局し、適切な表示レイアウトで自動的に表示されるので、容易にこれらの番組を視聴することができる。

【0062】更に、本形態では、視聴を行うテレビ受信機が同時に使用できるチューナの数を調べ、テレビ受信機に実行可能な可能な、あるいはテレビ受信機に適切な表示レイアウトが選択されて実行され、また、表示レイアウトに即したユーザ操作インターフェースが提供されるので、受信機に応じた適切な視聴形態をとることが可能となる。

【0063】また、こうしたデータ放送番組は、データ放送番組を企画する放送局や放送局に番組提供する者に

より、他の放送局の番組放送予定を基に独自の番組企画に基づいて作成され、この企画に基づいた同時視聴番組の組み合わせが選ばれ、企画に沿った表示レイアウト、操作インターフェースと共に、データ放送番組としての広告など企画に沿った表示要素を組み込むことが可能となる。同時にそれらの表示は視聴者が使用するテレビ受信機の能力に基づいた適切な表示レイアウトで行うことが可能となる。

【0064】以上に説明した実施例においては、テレビ受信機の能力を使用可能なチューナーの数で判断して、これに応じた動作を行うデータ放送、およびテレビ受信機のシステムを説明したが、データ放送の動作に影響を与えるテレビ受信機の能力、あるいは属性はチューナーの数に限ったものではない。例えば、チューナーに接続されるデコーダーの能力を調べ、一つのチューナーから同時に視聴できる番組の数によってレイアウトや操作系を変更することや、オーディオレコーダーの能力を調べ、複数の番組の音声出力の混合比を制御することなども可能である。

【0065】また、本形態においては表示部を含むテレビ受信機に対して本発明を適用したが、表示部をその構成に含まない、いわゆるセットトップボックスの構成をとることも可能である。その場合、合成プロセッサは接続されたディスプレイで表示するための映像表示信号を出力する機能を含む。その他の構成要素の機能、動作は以上に説明した通りである。

【0066】次に、本発明の第2の実施形態について説明する。

【0067】図19は本発明の第2の実施例におけるマルチ画面テレビ視聴システムの構成を示した図である。

【0068】図19において、1901は地上波テレビ放送受信アンテナ、1902、1903はアンテナ1901により受信された信号から指定されたチャンネルを選局し、検波を行ってビデオ信号とオーディオ信号を出力する機能を含むアナログテレビチューナー、1904、1905はNTSC方式によるコンポジットビデオ信号から輝度信号、色差信号と同期信号を分離し、クロックを生成して、輝度信号、色差信号をA/D変換する機能を含むNTSCプロセッサ、1908、1909はNTSCプロセッサ1904、1905から出力されるデジタルビデオ信号の解像度を表示するサイズに合わせて変換する解像度変換部、1910は入力される二つのデジタルビデオ信号と後述のグラフィック生成部1911で出力されるビデオ信号とを合成して表示部1912で表示するための映像信号を生成する合成プロセッサ、1912は合成プロセッサで合成された映像を表示する表示部、1906はチューナー1902、1903から出力されるオーディオ信号のバランス、音量、音質などを調整して増幅して出力するオーディオアンプ、1907はオーディオアンプからの出力される増幅された音声信号に基づいて音声を出力するスピーカである。

【0069】また、1913はテレビ受信機の操作ボタンとリモコンとを含む操作入力部、1915は図示しない電話回線を経由してネットワーク1924に接続される、あるいは直接ネットワークに接続されるネットワークインターフェース、1914は以上に説明したテレビ受信機1900の各部を制御しつつ、操作入力1913から入力されるユーザ操作情報とネットワークインターフェース1915から入力される各種情報に基づいて、テレビ表示に必要な文字図形をグラフィック生成部1911に描画させる機能を有する制御部である。

【0070】また、図19の1921、1922はテレビ放送の放送局であり、例えば、図20に示すような予め決められた放送番組予定表に従って放送番組を配信する。1923は後で説明するマルチ画面テンプレート機能を果すテレビ受信機のダウンロードアプリケーションをインターネット1924を経由して配給するマルチ画面テンプレートサービス提供者である。マルチ画面テンプレートサービス提供者1923は、放送局1921、1922が放送する放送番組の放送予定を入手し、これらの放送の選局制御と受信機のマルチ画面機能制御を行うテレビアプリケーションを作成して、視聴者のテレビ受信機へダウンロードさせるサービスを提供する。

【0071】まず、テレビジョン受信機1900の動作について説明する。

【0072】制御部1914の制御に基づいてチューナー1902によって受信されたビデオ信号は、NTSCプロセッサ1904でデジタル信号に変換され、解像度変換部1908で適当な解像度変換を経た後、合成プロセッサ1910により表示部1912に表示される。また、チューナー1902によって受信されたオーディオ信号はオーディオアンプ1908で調整、増幅されてスピーカ1907から出力される。この一連の動作により、選局した放送局が放送する一つの放送番組を視聴することが出来る。

【0073】またテレビ受信機1900は、チューナー1903からNTSCプロセッサ1905、解像度変換部1909を経る第2の受信系を持ち、合成プロセッサ1910における表示映像の切り替えと、オーディオアンプ1906における入力の切り替えにより、チューナー1903で受信した別の一つの放送番組を視聴することが出来る。

【0074】また、解像度変換部1908、1909における適当な解像度変換と、合成プロセッサ1910の合成処理により、チューナー1902とチューナー1903で各々受信した二つの放送番組を表示部1912に同時に表示して視聴することができる。このとき、オーディオアンプ1906では視聴者の操作に基づいて、一方の放送番組の音声を出力する。図21はこうした2画面表示状態を示すものである。

【0075】図22は、制御部1914に含まれるCP

Uで動作するソフトウェアの構成を示した図である。テレビアプリケーションは、操作入出力ドライバを介して得られるユーザ操作情報に基づいて、受信制御ミドルウェアおよび合成制御ミドルウェアに用意されたAPIを利用してテレビ受信機の動作を制御する。APIの呼び出しに基づきオペレーティングシステムの制御下でチューナドライバおよび合成プロセッサドライバを介して、前述の通りテレビ受信機の制御が行われる。

【0076】またテレビアプリケーションはグラフィックライブラリのAPIを通じ、グラフィックドライバを介して文字、図形、イメージなどをグラフィック生成部に描画させ、合成プロセッサによって合成して画面に表示する。この動作により画面上に受信した放送のチャンネルや、音量調整状態などの情報が画面表示を通じてユーザに通知される。

【0077】テレビアプリケーションは、テレビ受信機1900の制御部1914に含まれる記憶部に予め格納されたプログラムとして存在しテレビ受信機1900で実行される組み込みプログラムだけではなく、ネットワークインターフェース1915を通じてネットワーク上のサーバなどからダウンロードされ、制御部1914の記憶部に格納されて実行されるダウンロードプログラムとがある。

【0078】図23は図19に示したテレビアプリケーションのダウンロードサービスを提供するサービス提供者1923におけるダウンロードプログラムの作成と、ダウンロードの手順を説明する図である。

【0079】図23はテレビ受信機1900とサービス提供者1924のサービス提供サイトとの間のネットワークアクセスを中心に、両者における動作と手順を記述している。図23の手順のうち下線部は視聴者であるテレビ受信機1900のユーザによる操作である。

【0080】初めにユーザはテレビ受信機1900に備えられたテレビアプリケーションとしてのウェブブラウザの如きネットワークアクセス機能によってサービス提供サイトの視聴条件設定ページにアクセスする(S2301)。サービス提供サイトでは要求に従って視聴条件入力フォームを送信し(S2302)、テレビ受信機1900の表示部1912に設定ページとして表示される(S2303)。

【0081】図24は視聴条件設定ページとして表示部1912に表示される画面を示す図である。表示された画面は、サービス提供サイトで作成されるマルチ画面テンプレートの作成条件を設定する視聴条件入力フォームとなっている。ユーザは表示される質問に答える形でフォームを埋める。図24の入力結果によれば、朝の出勤時間前の忙しい時間にユーザが希望する情報が得られるテレビ番組を自動的に選局し、マルチ画面のレイアウトを自動的に制御して表示するマルチ画面テンプレート機能が提供される。

【0082】この機能は、サービス提供者が企画して提供するもので、ユーザの希望する情報の指定とユーザの受信機に関する情報をユーザプロファイルとして収集し、ユーザプロファイルに基づいて作成されダウンロードプログラムとしてテレビ受信機にダウンロードされる。

【0083】図24の例では、ユーザ使用するテレビ受信機において同時に使用できるチューナの数と、マルチ画面テンプレートの動作時間の設定、ユーザが希望する情報の種類をテレビ番組の情報ジャンルという形で指定する(S2304)。画面の「登録」ボタンを選択すると、制御部1914は設定された情報をユーザプロファイルとしてテレビ受信機のユーザ情報記憶部1916に格納する(S2305)。

【0084】次にユーザは、サービス提供サイトのマルチ画面テンプレートダウンロードページにアクセスする(S2306)と、サービス提供サイトはユーザプロファイルを要求する動作を含むページを送信し(S2307)、テレビ受信機1900のブラウザにより表示部1912に表示する(S2308)。そして、ユーザは表示されたページのダウンロード開始ボタンを選択することによりダウンロードを要求し(S2309)、ブラウザによってユーザプロファイルがユーザ情報記憶部1916から読み出されサービス提供サイトへ送信される(S2310, S2311)。

【0085】サービス提供サイトでは、受信したユーザプロファイルと、放送局1921, 1922の放送番組予定情報に基づいてマルチ画面テンプレートの機能を果すテレビアプリケーションプログラムを作成する(S2312, S2313)。このとき、サービス提供者が立てた企画に沿って、あるいはまた受信したユーザプロファイルに基づいて、サービス提供者独自の情報や、表示要素が追加されることがある。作成されたテレビアプリケーションはサービス提供者1923よりダウンロードプログラムとしてテレビ受信機1900へ送信される(S2314)。送信されたダウンロードプログラムはテレビ受信機1900により受信され(S2315)、制御部1914内の記憶部に格納され、実行される(S2316)。

【0086】ユーザプロファイルデータはユーザ情報記憶部1916に格納されるので、一旦作成されたユーザプロファイルは、内容に変更を必要とする場合を除いて、自動的に反復利用される。ユーザはマルチ画面テンプレートのダウンロードページのアクセス(S2306)以降の操作を行うだけで、最新の番組放送予定に基づいたマルチ画面テンプレートをダウンロードすることができる。

【0087】図25、図26は、ダウンロードされたダウンロードプログラムのソースコードの例を示すものである。図25、図26で一つのプログラムを示してい

る。プログラムはJAVA（登録商標）言語を用いて記述されたMorningInformationクラスであり、親クラスTvAppletクラスを継承したものである。TvAppletクラスは、ダウンロードプログラムがアプリケーション実行管理下で実行するために必要な機能を備えたクラスである。即ち、TvAppletクラスにおいてはinit()、start()、stop()、destroy()メソッドを持ち、アプリケーション実行管理がTvAppletクラスオブジェクトをロードした時にinit()メソッドを、実行を開始するときにstart()メソッドを、実行を停止するときにstop()メソッドを、プログラムをアンロードして破棄するときにdestroy()メソッドを呼ぶので、TvAppletクラスの継承クラス（子クラス）はその時点での必要な動作をこのメソッドにオーバーライドして記述する。

【0088】また、setAlarm()メソッドを備え、オペレーティングシステムが管理するシステム時刻がsetAlarm()メソッドで指定した時刻と一致したときonAlarm()メソッドを呼び出す機能を有する。TvAppletクラスの継承クラス（子クラス）では指定時刻に実行したい制御や動作をこのメソッドにオーバーライドして記述する。更に、操作入力部1913を通じてユーザの操作情報が入力されたとき、onUserOperate()メソッドが呼ばれるので、TvAppletクラスの継承クラス（子クラス）はユーザ入力を解析して、必要な動作を行えるようこのメソッドをオーバーライドすればよい。

【0089】また、テレビアプリケーションプログラムはTunerクラスを利用することができる。Tunerクラスはテレビ受信機1900の受信動作をオブジェクト化したクラスで、setTune()メソッドにより選局するチャンネルを指定し、setDisplaySize()メソッドにより表示解像度を指定し、setPriority()メソッドにより他の表示オブジェクトとの表示優先度を指定することができる。capture()メソッドは、Tunerオブジェクトがその時点で表示している映像をキャプチャして、その静止画像をImageオブジェクトとして返す。またsetAudioMode()メソッドによりオーディオアンプ1906の制御を行うことができる。即ち、引数により消音（0）、主音声出力（1）、副音声出力（2）、主+副音声出力（3）が指定される。複数のチューナーから出力の指定が同時になされた場合は、両者がミックスされる。また、クラスメソッドopen()メソッドを備え、システムが可能であればTunerクラスのインスタンスを生成して参照を与えてくれる。

【0090】TunerオブジェクトはTvAppletオブジェクトが持つグラフィックオブジェクトTvScreenに描画されることによって画面に映像が表示される。即ちTvScreenクラスのdrawImageメソッドで、描画対象としてのTunerクラスのインスタンスと、これを描画する位置とを指定する。TvAppletでは画面表示を行うときpaint()メソッドが呼ばれるので、paint()メソッドにおいて表示したいチューナーと、他のグラフィック要素を指定して描画す

る。

【0091】図25、図26においてテレビアプリケーションMorningInformationクラスの動作を説明するが、放送局1921、1922において、図20で示す放送番組予定表に従った放送番組が放送されていることが前提となる。図25、図26のプログラムにより、この放送番組予定を前提として、時間の経過と共にチャンネルを切り替え、複数の番組をマルチ画面として合成して視聴する動作が行われる。

【0092】図25において、テレビアプリケーションMorningInformationクラスのメソッドinit()ではプログラムがロードされたときの初期化を行っている。即ち、チューナのインスタンスtv0を生成し、1チャンネルを選局し、表示解像度を480×360に、表示優先度を0（最優先）に、音声出力を消音状態に指定している。また、もう一つのチューナインスタンスtv1を生成し、4チャンネルを選局し、表示解像度を120×90に、表示優先度を2に、音声出力を消音状態に指定している。最後に状態変数stageを0とし、時刻7時45分にcallbackを指定している。

【0093】メソッドstart()においては、init()メソッドで初期化された状態でプログラムの実行がスタートしたとき、tv1チューナの音声が outputされるよう指定している。実際に出力される音量は、このダウンロードプログラムの外側でユーザ操作により決定された音量指定に従う音量である。同時にpaint()メソッドが呼ばれ、状態変数stageの値0に従ってメソッドlayout0()で指定された描画が行われる。

【0094】即ち、後で説明するようにアプリケーション実行管理がダウンロードプログラムを時刻7時30分に起動したとき、チャンネル1の番組「天気予報」とチャンネル4の番組「ノンストップ世界旅行」とサービス提供者1923が表示する独自の広告AD21が図27の如く合成表示され、番組「ノンストップ世界旅行」の音声が outputされる。

【0095】時刻が7時45分となったときメソッドonAlarm()が呼ばれる。引数timeは現在時刻を示しているので、if文による分岐に従いtv0の選局がチャンネル2に変更され、時刻による次のcallbackを7時54分に指定する。即ち、図27に示す番組「天気予報」の表示は番組終了とともにチャンネル2で放送中の番組「各地の話題」に切り替わる。

【0096】時刻が7時54分となったときメソッドonAlarm()が呼ばれる。if文の制御に従い、tv1の選局がチャンネル3に変更され、時刻による次のcallbackを7時55分に指定する。同時に状態変数stageの値を1に変更する。即ち、図28に示す通り、番組「世界旅行」の表示に代わって、位置と大きさの違う番組「産業ニュース」とサービス提供者1923が表示する独自の広告AD22、AD23の表示が行われる。

【0097】時刻が7時55分となったときメソッドonAlarm()が呼ばれる。if文の制御に従い、tv1の表示がキャプチャされ、その静止画イメージがhighlightに格納される。tv1の選局がチャンネル4に変更され、時刻による次のcallbackを7時55分に指定し、状態変数stageの値を2に変更する。即ち、図29に示す通り、番組「産業ニュース」の中で7時55分に放送された今日のニュースの放送画面がキャプチャされた静止画として表示され、他にチャンネル2の番組「交通情報」とチャンネル4の番組「世界旅行」が受信され、合成されて表示される。

【0098】時刻が8時00分となったとき、メソッドonAlarm()が呼ばれる。if文の制御に従い、tv0の選局がチャンネル1に変更される。即ち、図29において番組「交通情報」は番組の終了後、チャンネル1の番組「ニュース」に切り替わって表示される他、サービス提供者1523が表示する独自の広告AD24が表示される。

【0099】本形態においては、実行管理情報ファイルがダウンロードプログラムと一緒にダウンロードされる。実行管理情報ファイルには、ダウンロードプログラムの名称及びファイル名の情報、ダウンロードプログラムの動作を説明する文章を構成するテキスト、ダウンロードプログラムの動作を開始する日付及び時間情報、ダウンロードプログラムの動作を終了する日付及び時間情報、などが含まれている。

【0100】図22のアプリケーション実行管理では、ダウンロードプログラムと実行管理情報ファイルをダウンロードし、ダウンロードプログラムを格納すると共に、実行管理情報ファイルのこれらの情報を集め、適切なタイミングで該当するダウンロードアプリケーションの実行を開始する。ダウンロードアプリケーション「MorningInformation」の場合は、前述の通り朝7時30分に実行が開始される。

【0101】以上の説明の通り、本形態によれば、ユーザーの希望とテレビ受信機の性能情報に基づくユーザプロファイルと、各放送局におけるテレビ番組の番組放送予定情報とに基づき、テレビアプリケーションダウンロードサービスの提供者がテレビアプリケーション「Morning Information」を作成して、ネットワークを介してテレビ受信機に配信（ダウンロード）している。

【0102】そして、テレビ受信機がこのダウンロードされたテレビアプリケーション「MorningInformation」を実行することで、テレビ受信機の複数の受信機能を制御し、複数の画面を合成表示するマルチ画面におけるレイアウト制御を行い、マルチ画面テンプレート機能を実現することができる。

【0103】本形態のマルチ画面テンプレート機能では、一例として、視聴者であるユーザーに対して出勤前の忙しい時間帯に必要な情報が漏れなく集められるように、複数の放送局から放送される番組を時間の経過と共に

順に切り替えて表示するものである。また、同時にリラックスした雰囲気を演出するために、環境ビデオ的な番組を選択してマルチ画面に表示している。

【0104】こうした番組は、通常、視聴者であるユーザーが新聞や雑誌、その他のメディアを通じて提供される放送番組予定表等を視聴の目的に照らして精査することによって選択、受信する必要があるが、本形態によれば、テレビ受信機が接続されたネットワーク上のサービス提供者が提供するサービス企画を選択して、目的に適ったプログラムをダウンロードするだけで目的にふさわしい番組を自動的に受信し、最適な表示レイアウトにて表示することが可能となる。

【0105】ダウンロードするプログラムはユーザーの視聴目的とユーザーが使用する受信機の性能を反映したユーザプロファイルに基づいて作成されるため、一旦作成されたユーザプロファイルは変更の必要がない限り自動的に反復利用することができ、毎回プロファイル設定を行うためにユーザーの手を煩わす必要はない。

【0106】また従来、高機能なマルチ画面のレイアウトの設定を、テレビ受信機のリモコンや操作ボタンを通じて一つずつ行っていた手間も、ダウンロードしたプログラムが自動実行するために不要となり、高度なテレビ受信機の機能を最大限に活用することが可能となって、結果として非常に簡便に、目的に適ったテレビ視聴を行うことができる。

【0107】尚、本実施例においては、ダウンロードプログラムの実行開始、実行終了に関する情報をプログラムと同時にダウンロードして、テレビ受信機のアプリケーション実行管理がダウンロードプログラムの実行開始、及び終了を制御するよう説明したが、実行時間管理が可能でTvScreenのような表示オブジェクトによって表示画面を占有しないクラスを持ち、ダウンロード後時刻に関わらずこのクラスからプログラムの実行を開始し、視聴開始時刻に実施例で説明したようなTvAppletクラスの子クラスのオブジェクトを起動するような構成をとることも可能である。この場合、実行の終了もダウンロードプログラム自身で制御する。いずれの場合も、ユーザーによる他の操作によって、実行中のダウンロードプログラムの動作を中止する事が、アプリケーション実行管理の機能として可能である。

【0108】また、本実施形態では、表示部を含むいわゆる一体型テレビ受信機に対して本発明を適用した場合について説明したが、表示部をその構成に含まず、外部に接続されたディスプレイに映像表示信号出力の機能を含む合成プロセッサの出力によって映像の表示を行う、いわゆるセットトップボックスの構成をとることも可能である。

【0109】さらに、テレビ受信機のチューナはアナログチューナであるとして説明したが、デジタル放送を受信するデジタルチューナであってもよく、デジタル放送

によるデータ放送とは別に、ネットワークなどを経由して受信したプログラムによる制御を行うことが可能である。また、マルチ画面テンプレートがチューナを制御して受信する放送番組は、デジタル放送によるデータ放送番組でもよく、あるいはデータ放送を通じて放送される表示データ、実行可能なプログラムであってもよい。

【0110】

【発明の効果】以上説明したように、本発明によれば、視聴目的に適う複数の番組が自動的に選局され、番組の組み合わせとその内容に適した表示レイアウトになるようにテレビ受信機が自動的に制御される。視聴者であるユーザは、多くの番組の中から目的に適う番組を探し出す手間から開放され、また複数の画面ごとに選局と画面レイアウトを指定する操作から開放され、マルチ画面機能を有するテレビ受信機に高度な機能をより簡単に利用することが可能となる。

【図面の簡単な説明】

【図1】本発明が適用されるテレビジョン受信機の構成を示す図である。

【図2】受信動作に関する制御プログラムの様子を示す図である。

【図3】XMLデータの受信動作を説明するためのフローチャートである。

【図4】マルチ画面制御に係るXMLデータの様子を示す図である。

【図5】マルチ画面制御に係るXMLデータの様子を示す図である。

【図6】マルチ画面制御に係るXMLデータの様子を示す図である。

【図7】マルチ画面制御に係るXMLデータの様子を示す図である。

【図8】受信動作を説明するためのフローチャートである。

【図9】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図10】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図11】マルチ画面制御に係るXMLデータの様子を示す図である。

【図12】マルチ画面制御に係るXMLデータの様子を示す図である。

【図13】マルチ画面制御に係るXMLデータの様子を示す図である。

【図14】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図15】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図16】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図17】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図18】電子番組ガイドの様子を示す図である。

【図19】本発明が適用されるテレビジョン受信システムの構成を示す図である。

【図20】放送番組の様子を示す図である。

【図21】2画面同時表示モードの際の表示画面の様子を示す図である。

【図22】受信動作に関する制御プログラムを示す図である。

【図23】制御プログラムのダウンロード処理を示す図である。

【図24】ユーザプロファイルの設定画面を示す図である。

【図25】ダウンロードされる制御プログラムを示す図である。

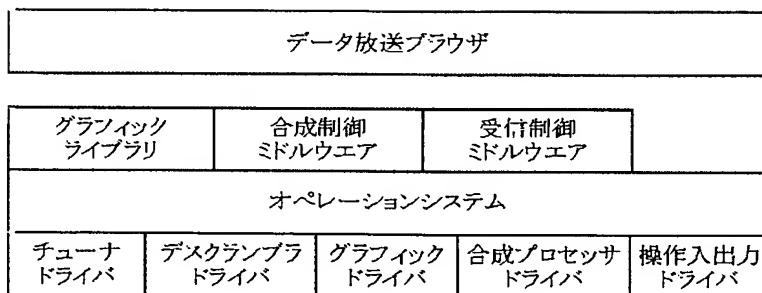
【図26】ダウンロードされる制御プログラムを示す図である。

【図27】マルチ画面テンプレート機能による表示画面の様子を示す図である。

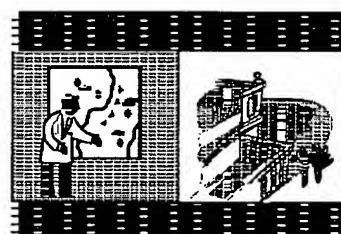
【図28】マルチ画面テンプレート機能による表示画面の様子を示す図である。

【図29】マルチ画面テンプレート機能による表示画面の様子を示す図である。

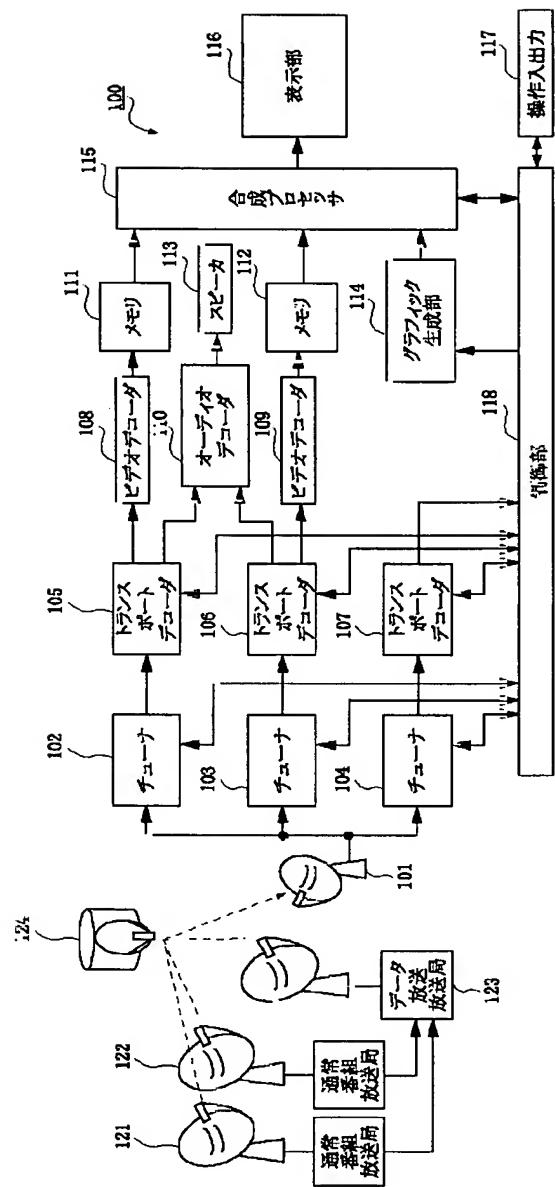
【図2】



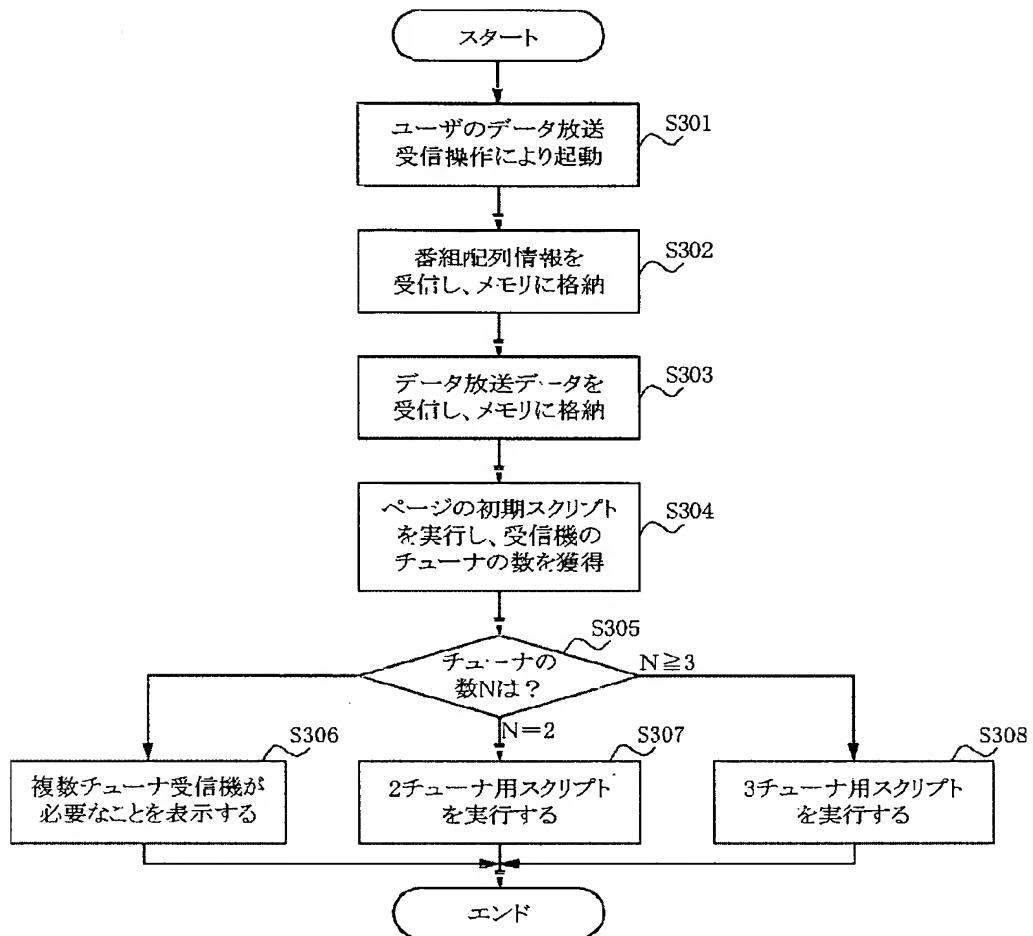
【図21】



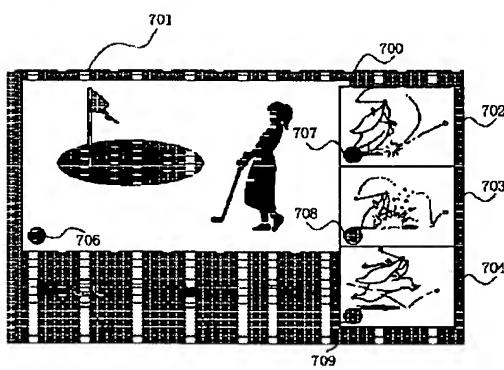
【図1】



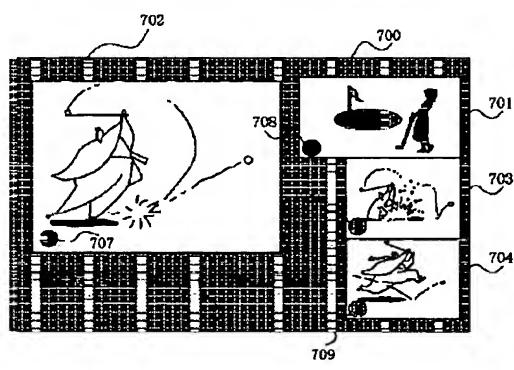
【図3】



【図9】



【図10】



【図4】

```

<!DOCTYPE dbc SYSTEM "http://www.xxx.co.jpdbc.dtd">
<head>
<title>PARADISE FOR GOLF FREAKS</title>
<script language="JavaScript">

function initialize(){
    var n = Browser.numberTuner();
    switch( n ){
        case 0:
        case 1:
            // show alert according to the body part in this document
            break;
        case 2:
            document.open("golfParadiseBy2Tuners.xml");
            break;
        case 3:
            document.open("golfParadiseBy3Tuners.xml");
            break;
        default:
            document.open("golfParadiseBy3Tuners.xml");
    }
}

</script>
</head>

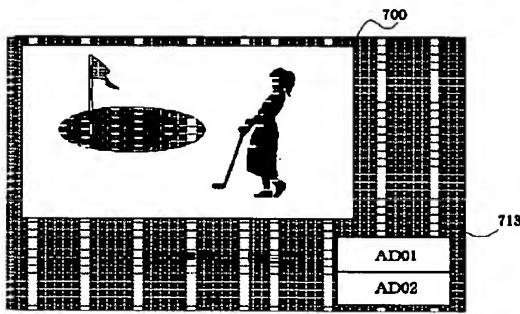
<body onLoad="initialize()">

<div style="position:absolute; top:800;left:1760; width:136; height:64; visibility:visible;">
この番組はマルチチューナーテレビでお楽しみください
</div> } 401

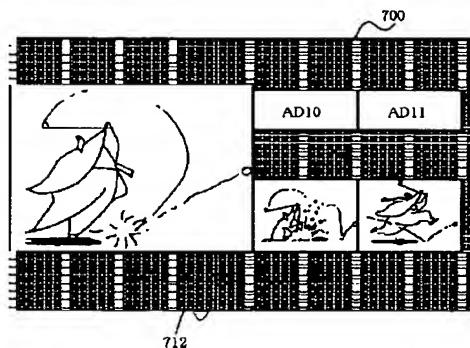
</body>

```

【図14】



【図15】



【図22】

テレビアプリケーション				
JVM				
アプリケーション実行管理	グラフィックライブラリ	合成制御ミドルウェア	受信制御ミドルウェア	
オペレーションシステム				
ネットワークドライバ	チューナードライバ	グラフィックドライバ	合成プロセッサドライバ	操作入出力ドライバ

【図5】

```

<!DOCTYPE dbc SYSTEM "http://www.xxx.co.jpdbc.dtd">
<head>
<title>PARADISE FOR GOLF FREAKS</title>
<script language="JavaScript">

var tuner0, tuner1; //チューナオブジェクト
var source0, source1, source2, source3; //サービスオブジェクト
var service0, service1, service2, service3; //サービスオブジェクトID
var pos0, pos1, pos2, pos3; //サービスが置かれている位置
var central; //中央位置に置かれているサービス
var btnR, btnG, btnB, btnY; //ボタンオブジェクト

function initialize(){
    tuner0 = Browser.openTuner();
    source0 = tuner0.addService("broadcasterA", "101");
    service0 = document.getElementById("video0");
    btnR = document.getElementById("buttonRed");

    tuner1 = Browser.openTuner();
    source1 = tuner1.addService("broadcasterB", "161");
    service1 = document.getElementById("video1");
    btnG = document.getElementById("buttonGreen");

    source2 = tuner1.addService("162");
    service2 = document.getElementById("video2");
    btnB = document.getElementById("buttonBlue");

    source3 = tuner1.addService("163");
    service3 = document.getElementById("video3");
    btnY = document.getElementById("buttonYellow");

    setPosition0( 0 );
    setPosition1( 1 );
    setPosition2( 2 );
    setPosition3( 3 );
}

function setPosition0( n ){
    switch( n ){
        case 0:
            service0.style = "position:absolute; top:20; left:40; width:960; height:540; visibility:visible;";
            btnR.style = "position:absolute; top:860; left:70; width:60; height:60; visibility:visible;";
            document.all.audio0.style="sonority:sonorous";
            break;
        case 1:
            service1.style = "position:absolute; top:40; left:40; width:800; height:600; visibility:visible;";
            btnG.style = "position:absolute; top:520; left:70; width:60; height:60; visibility:visible;";
            document.all.audio1.style="sonority:sonorous";
            break;
        case 2:
            service2.style = "position:absolute; top:40; left:40; width:800; height:600; visibility:visible;";
            btnB.style = "position:absolute; top:520; left:70; width:60; height:60; visibility:visible;";
            document.all.audio2.style="sonority:sonorous";
            break;
        case 3:
            service3.style = "position:absolute; top:40; left:40; width:800; height:600; visibility:visible;";
            btnY.style = "position:absolute; top:520; left:70; width:60; height:60; visibility:visible;";
            document.all.audio3.style="sonority:sonorous";
            break;
    }
}

function setPosition1( n ){
    ....
}

function setPosition2( n ){
    ....
}

function setPosition3( n ){
    ....
}

```

【図6】

```

function removal( pos ){
    //あるサービスの現在位置がposのとき(switch文)
    //posがiならば、サービスcentralを位置iへ立ち退かせる
    //サービスcentralの位置に新しい位置iを代入する
    switch( pos ){
        case 0:
            setPosition0( central );
            newPos = 0;
            break;
        case 1:
            setPosition1( central );
            newPos = 1;
            break;
        case 2:
            setPosition2( central );
            newPos = 2;
            break;
        case 3:
            setPosition3( central );
            newPos = 3;
            break;
    }
    switch( central ){
        case 0: pos0 = newPos; break;
        case 1: pos1 = newPos; break;
        case 2: pos2 = newPos; break;
        case 3: pos3 = newPos; break;
    }
}

function selectRed0(){
    //サービス番号centralとサービス0の位置を交換する
    removal( pos0 );
    setPosition0( 0 );
}

function selectGreen(){
    //サービス番号centralとサービス1の位置を交換する
    removal( pos1 );
    setPosition0( 1 );
}

function selectBlue(){
    //サービス番号centralとサービス2の位置を交換する
    removal( pos2 );
    setPosition0( 2 );
}

function selectYellow(){
    //サービス番号centralとサービス3の位置を交換する
    removal( pos3 );
    setPosition0( 3 );
}

</script>
</head>

```

【図7】

```

<body onLoad="initialize()"
  onButtonRed="selectRed()"
  onButtonGreen="selectGreen()"
  onButtonBlue="selectBlue()"
  onButtonYellow="selectYellow()">
<div id = "div_video0">
  <object id="video0", type="video/mpeg2", source=source0></object>
</div>
<div id = "div_video1">
  <object id="video1", type="video/mpeg2", source=source1></object>
</div>
<div id = "div_video2">
  <object id="video2", type="video/mpeg2", source=source2></object>
</div>
<div id = "div_video3">
  <object id="video3", type="video/mpeg2", source=source3></object>
</div>

<div id = "div_audio0">
  <object id="audio0", type="audio/mpeg2", source=source0></object>
</div>
<div id = "div_audiol">
  <object id="audiol", type="audio/mpeg2", source=source1></object>
</div>
<div id = "div_audio2">
  <object id="audio2", type="audio/mpeg2", source=source2></object>
</div>
<div id = "div_audio">

  <object id="audio3", type="audio/mpeg2", source=source3></object>
</div>

<div style="position:absolute; top:580; left:60; width:960; height:80;">
  <object id="advertisement0", type="image/jpeg" data="ad0.jpg"></object>
</div>

<div style="position:absolute; top:680; left:60; width:300; height:60;">
  <object id="fourButtons", type="image/jpeg" data="fourBuddons.jpg"></object>
</div>
<div style="position:absolute; top:680; left:360; width:500; height:60;">
  ボタンで画面が拡大されます
</div>

<div id = "div_bntR">
  <object id="buttonRed", type="image/jpeg" data="bntR.jpg"></object>
</div>
<div id = "div_bntG">
  <object id="buttonGreen", type="image/jpeg" data="bntG.jpg"></object>
</div>
<div id = "div_bntB">
  <object id="buttonBlue", type="image/jpeg" data="bntB.jpg"></object>
</div>
<div id = "div_bntY">
  <object id="buttonYellow", type="image/jpeg" data="bntY.jpg"></object>
</div>
</body>

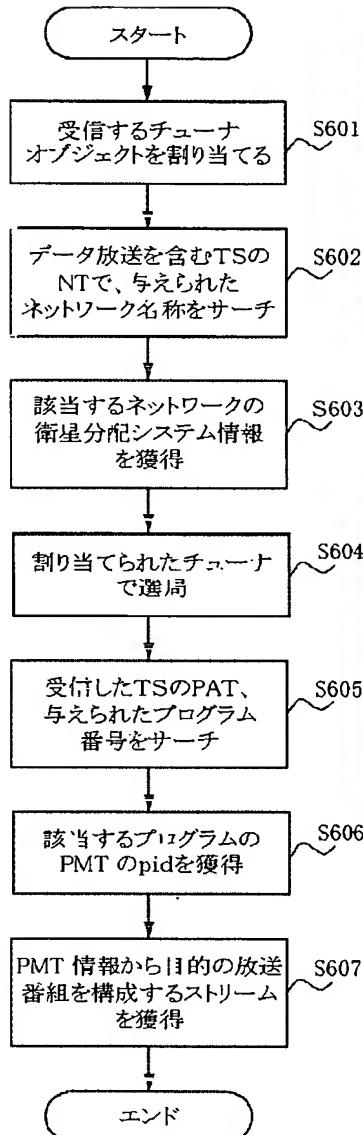
```

501

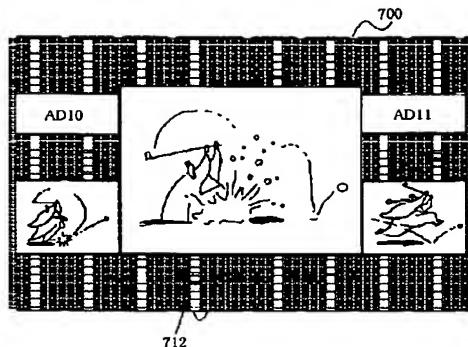
502

503

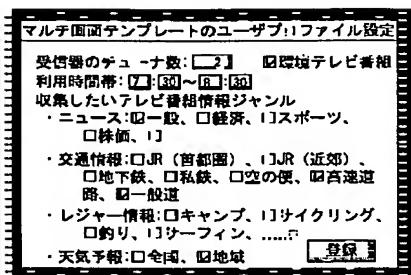
【図8】



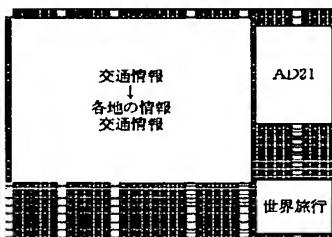
【図16】



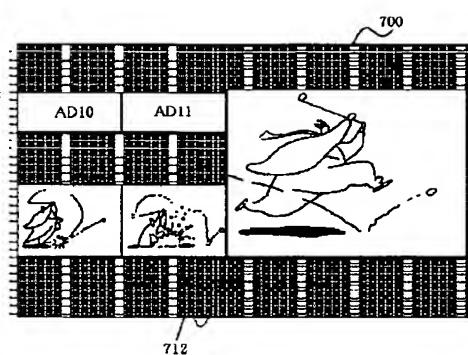
【図24】



【図27】



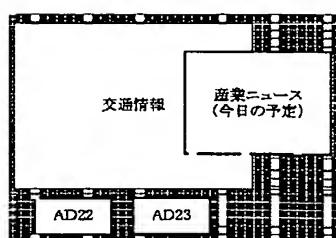
【図17】



【図18】

	101ch	161ch	162ch	163ch	421ch
12:00--	Music		ヨットレース中継		観戦ツアー案内
13:00--		World News	男子プロゴルフツアー中継		
14:00--			16H 17H 18H		極楽ゴルフ三昧
15:00--	女子プロゴルフツアー中継				
16:00--			競馬中継		ゴルフ教室
17:00--	Moive				

【図28】



【図11】

```

<!DOCTYPE dbc SYSTEM "http://www.xxx.co.jpdbc.dtd">
<head>
<title>Data PARADISE FOR GOLF FREAKS</title>
<script language="JavaScript">

var tuner0; //チューナオブジェクト
var source0, source1, source2, source3; //サービスオブジェクト
var service0, service1, service2, service3; //サービスオブジェクトID
var magnified; //拡大表示されているサービス
var previous0, previous1, previous2, previous3;

function initialize(){
    ...
}

function setPosition0( n ){
    if(previous0 = -1)
        tuner0.addService("broadcasterA", 101);
    if(n = -1)
        tuner0.clearService();
    else if(n : 0){
        service0.style="position:absolute; top=.....;";
        document.all.audio0.style="sonority:mute;";
        ad01.style="visibility:visible;";
        ad02.style="visibility:visible;";
    } else if(n = 1){
        service0.style="position:absolute; top=.....;";
        document.all.audio0.style="sonority:sonorous;";
        ad01.style="visibility:visible;";
        ad02.style="visibility:visible;";
    }
    previous0 = n;
}

function setPosition1( n ){
    if(previous1 = -1){
        tuner0.addService("broadcasterB", 161);
    if(n = -1)
        tuner0.removeService( service1 );
    else if(n : 0){
        service1.style="position:absolute; top=.....;";
        document.all.audio1.style="sonority:mute;";
        ad11.style="visibility:visible;";
    } else if(n = 1){
        ad11.style="visibility:invisible;";
        service1.style="position:absolute; top=.....;";
        document.all.audio1.style="sonority:sonorous;";
    }
    previous1 = n;
}

function setPosition2( n ){
    ....
}

function setPosition3( n ){
    ....
}

function selectLeft(){
    //より左の番組を拡大する
    switch( magnified ){
        case 0:
            break;
        case 1:
            setPosition1(-1);
    }
}

```

【図12】

```

        setPosition2(-1);
        setPosition3(-1);
        setPosition0(1);
        magnified =0;
        break;
    case 2:
        setPosition2( 0 );
        setPosition1( 1 );
        magnified = 1;
        break;
    case 3:
        setPosition3( 0 );
        setPosition2( 1 );
        magnified = 2;
        break;
    }
}

function selectRight(){
    //より右の番組を拡大する
    switch( magnified ){
        case 0:
            setPosition0(-1);
            setPosition1(1);
            setPosition2(0);
            setPosition3(0);
            magnified = 1;
            break;
        case 1:
            setPosition1( 0 );
            setPosition2( 1 );
            magnified = 2;
            break;
        case 2:
            setPosition1( 0 );
            setPosition2( 1 );
            magnified = 3;
            break;
        case 3:
            break;
    }
}

removal( pos1 ){
    setPosition0( 1 );
}

```

</script>

</head>

<body onLoad="initialize()" onButtonLeft="selectLeft()" onButtonRight="selectRight()">

```

<div id = "div_video0">
    <object id="video0", type="video/mpeg2", source=source0> </object>
</div>
<div id = "div_video1">
    <object id="video1", type="video/mpeg2", source=source1> </object>
</div>
<div id = "div_video2">
    <object id="video2", type="video/mpeg2", source=source2> </object>
</div>
<div id = "div_video3">
    <object id="video3", type="video/mpeg2", source=source3> </object>
</div>

```

【図13】

```

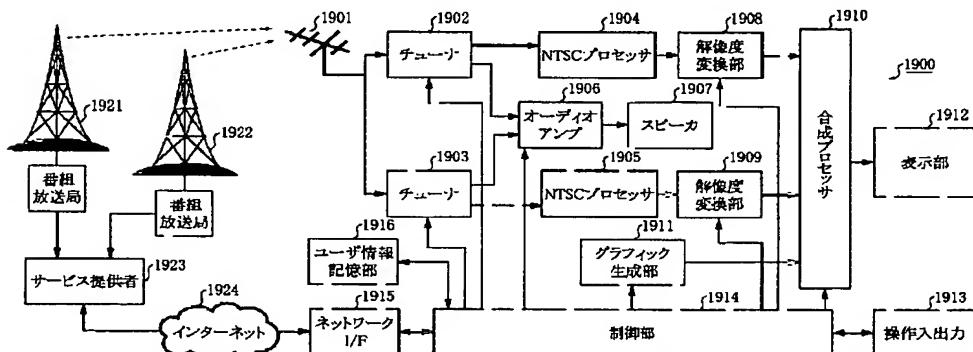
<div id = "div_audio0">
  <object id="audio0", type="audio/mpeg2", source=source0> </object>
</div>
<div id = "div_audio1">
  <object id="audio1", type="audio/mpeg2", source=source1> </object>
</div>
<div id = "div_audio2">
  <object id="audio2", type="audio/mpeg2", source=source2> </object>
</div>
<div id = "div_audio3">
  <object id="audio3", type="audio/mpeg2", source=source3> </object>
</div>

<div id = "div_ad01">
  <object id="advertisement01", type="image/jpeg" data="ad01.jpg"></object>
</div>
<div id = "div_ad02">
  <object id="advertisement02", type="image/jpeg" data="ad02.jpg"></object>
</div>
<div id = "div_ad11">
  <object id="advertisement11", type="image/jpeg" data="ad11.jpg"></object>
</div>
<div id = "div_ad12">
  <object id="advertisement12", type="image/jpeg" data="ad12.jpg"></object>
</div>

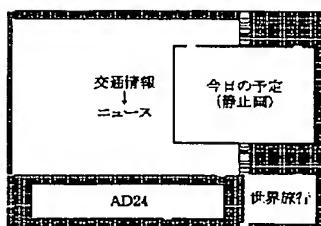
<div style="position:absolute; top:680; left:160; width:700; height:60;">
  ←,→ボタンで拡大画面が切り替わります
</div>
</body>

```

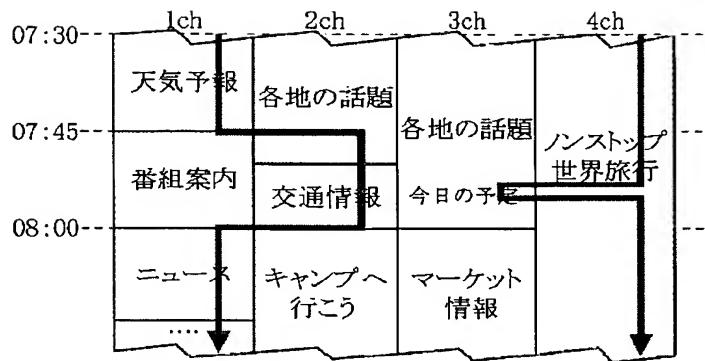
【図19】



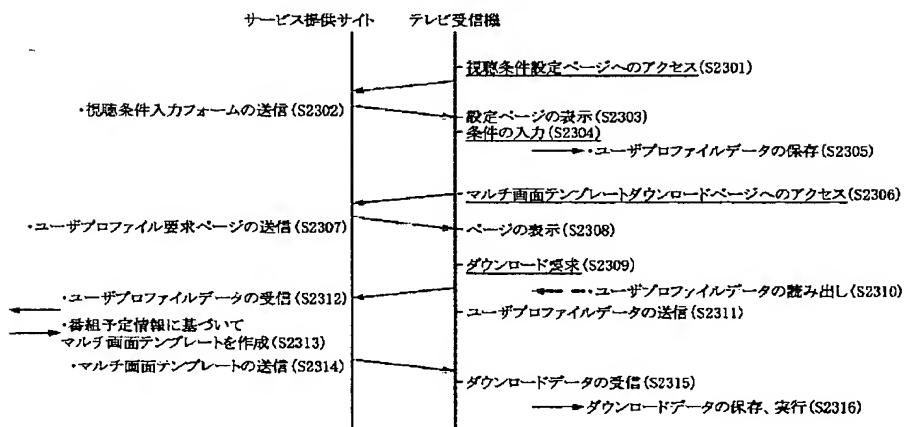
【図29】



【図20】



【図23】



【図25】

```

public class MorningInformation extends TvApplet {

    Tuner tv0, tv1;
    int stage;
    Image highlight;

    public void init() {
        tv0 = TvAnalogTuner.open();
        add( tuner0 );
        tv0.setTune( 1 );
        tv0.setDisplaySize(480,360);
        tv0.setPriority( 1 );
        tv0.setAudioMode( 0 );

        tv1 = TvAnalogTuner.open();
        add( tv1 );
        tv1.setTune( 4 );
        tv1.setDisplaySize(120,90);
        tv1.setPriority( 2 );
        tv1.setAudioMode( 0 );

        stage = 0;
        setAlarm("07:45");
    }

    public void start(){
        tv1.setAudioMode( 3 );
        validate();
        setVisible( true );
    }

    public void stop(){
        setVisible( false );
        tv0.close();
        tv1.close();
    }

    public destroy(){
        ....
    }

    private void layout0(TvScreen s){
        s.drawImage(tv1, 510, 380);
        s.drawImage(tv0, 20, 20);
        s.drawImage("ad21.jpg", 510, 60);
    }

    private void layout1(TvScreen s){
        s.drawImage(tv1, 510, 380);
        s.drawImage(tv1, 300, 180);
        s.drawImage("ad22.jpg", 100, 390);
        s.drawImage("ad23.jpg", 300, 390);
    }

    private void layout2(TvScreen s){
        s.drawImage(tv1, 510, 380);
        s.drawImage(highlite, 300, 180);
        s.drawImage(tv0, 20, 20);
        s.drawImage("ad24.jpg", 100, 390);
    }

    public void Paint(TvScreen s){
        switch( stage ){
            case 0: layout0( s ); break;
            case 1: layout1( s ); break;
            case 2: layout2( s ); break;
        }
    }
}

```

【図26】

```

        }

    onAlarm(String time){
        if( time.equals("07:45")){
            tv0.setTune( 2 );
            setAlarm("07:54");
            validate();
        } else if( time.equals("07:54")){
            tv1.setTune( 3 );
            stage= 1;
            setAlarm("07:55");
            validate();
        } else if( time.equals("07:55")){
            highlite = tv1.caputure();
            tv1.setTune( 4 );
            stage = 2;
            setAlarm("08:00");
            validate();
        } else if( time.equals("08:00")){
            tv0.setTune( 1 );
            validate();
        }
    }

    onUserOperate( .... ){
        ....
        ....
    }
}

```

フロントページの続き

(51)Int.Cl.7	識別記号	F I	(参考)
H 04N 7/20	630	H 04N 7/20	630

(72)発明者 吉川 輝樹	F ターム(参考)	5B069 CA03 CA13
東京都大田区下丸子3丁目30番2号キヤノ	5C025 CA01 CA02 CA03 CA06 CA09	
ン株式会社内	CA10 CA11 CB08 CB10 DA01	
(72)発明者 水留 敦	DA04 DA05	
東京都大田区下丸子3丁目30番2号キヤノ	5C064 DA01 DA07 DA10	
ン株式会社内		